120 — Input device

130 — Output/Display device

140 — Computer readable medium

100

102 — Processor

Memory

104

110 — Spec Software (includes GUI)

112 — Library (Specs and diagrams)

114 — User Data (specs and diagrams being refined)

116 — Resulting software or software design

Fig. 1

Allow the user to build software specifications and diagrams

Allow the user to start refine his specifications and diagrams

Allow the user to automatically generate code from his specifications and diagrams

202

204

206

Fig. 2(a)
Spec Software

Fig. 2(b)
Spec Software

Express Mai No. EV442675628 US
Application No. 09/665,179
Title: Method & Apparatus for Determining
Colimits of Hereditary Diagrams
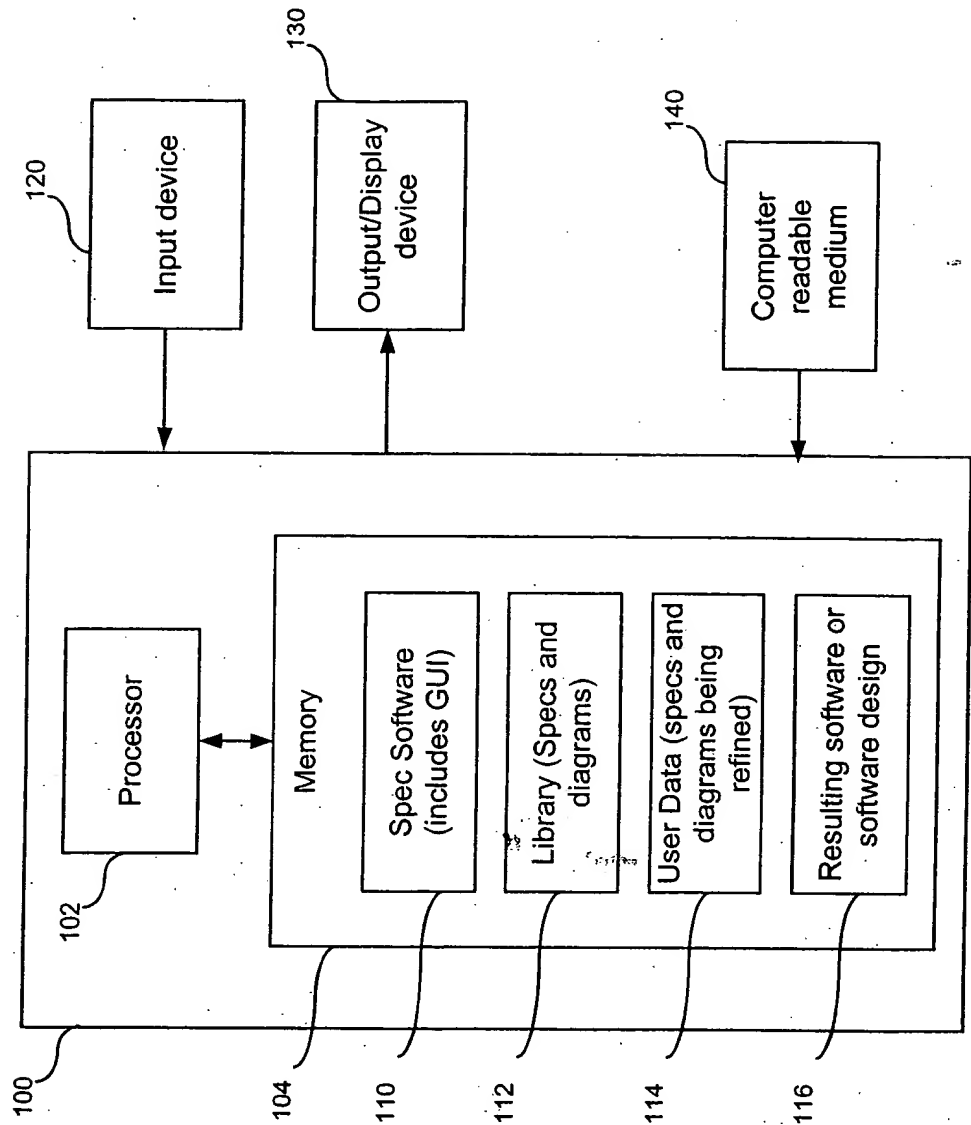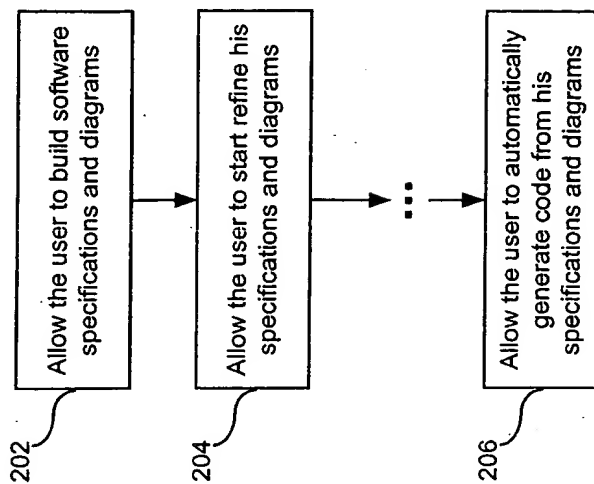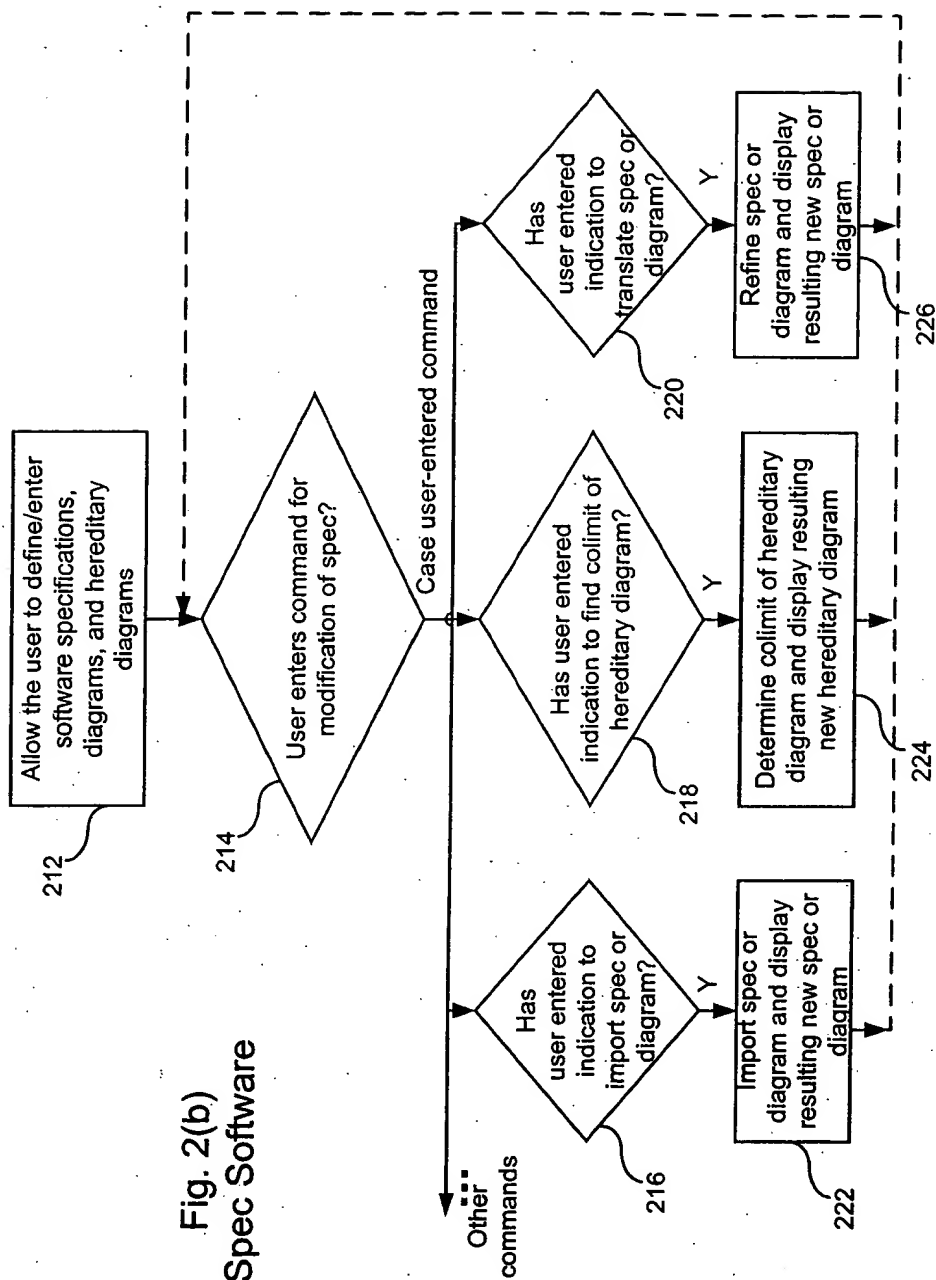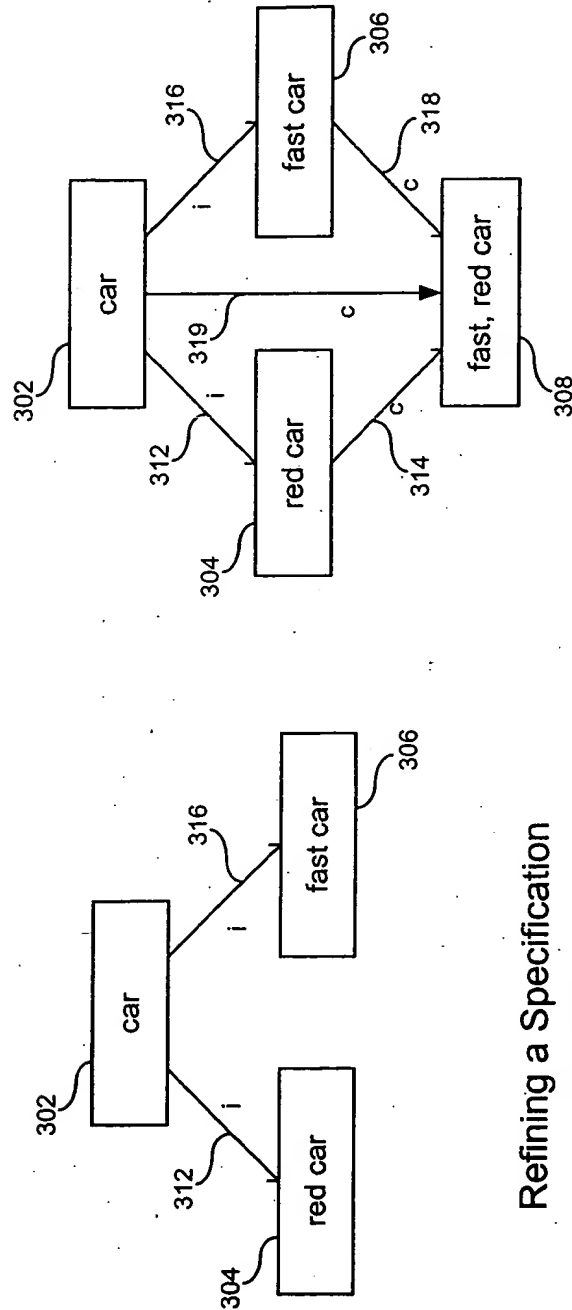Inventors: Dusko Pavlovic, et al.
Atty. Docket No.: 11128-04483

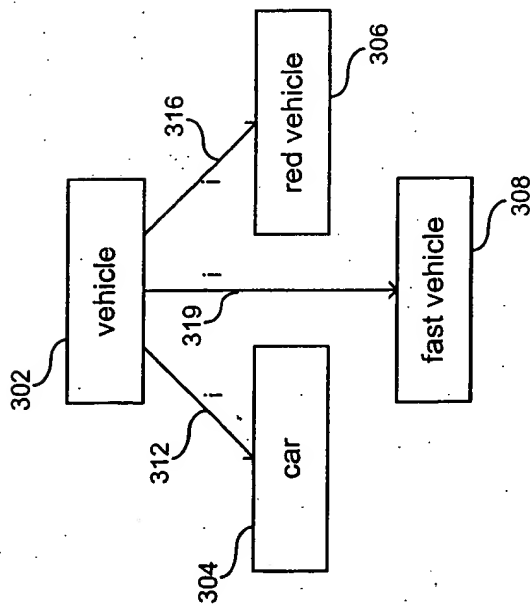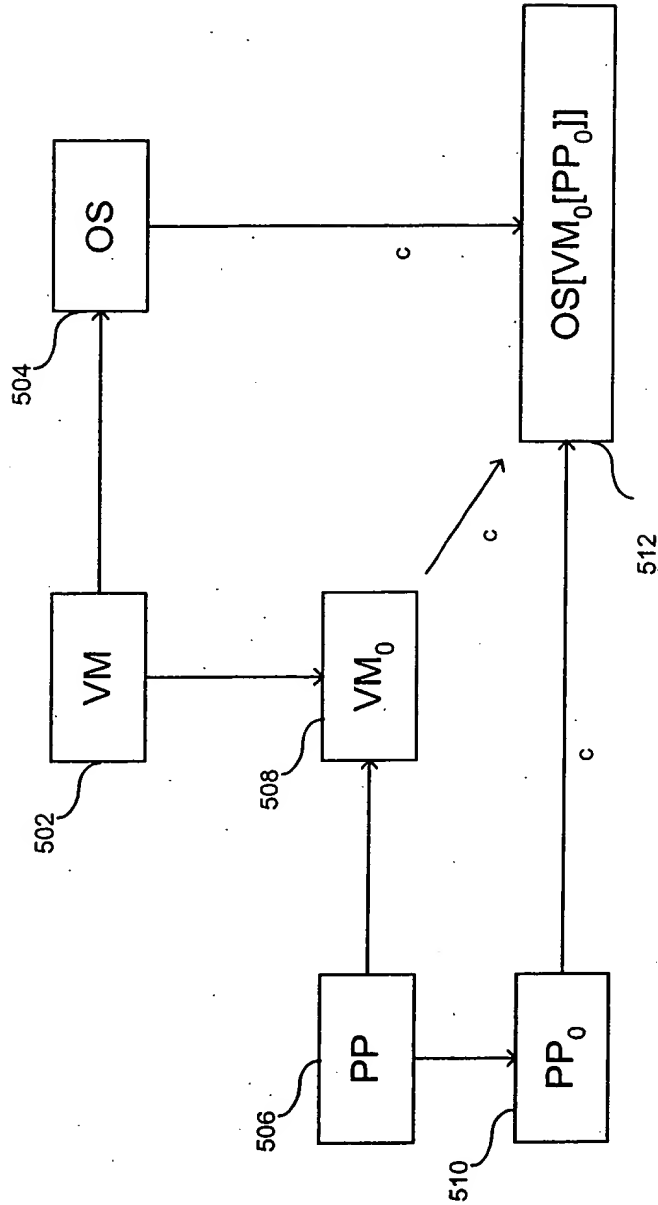Example of Using a Colimit to
Combine Refined Specifications
Fig. 3(b)

Refining a Specification
Fig. 3(a)

Example of Using a Colimit to
Combine Refined Specifications
Fig. 4(b)

Refining a Specification
Fig. 4(a)

Example Colimit of Specifications
Fig. 5
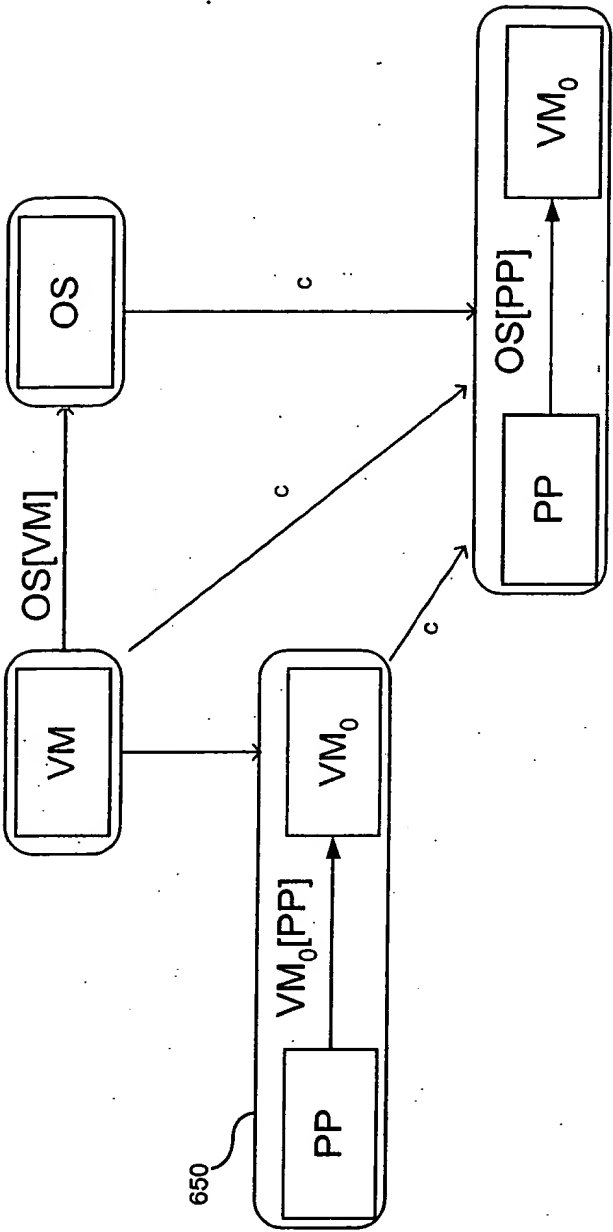
Example Colimit of Diagrams of Diagrams

Fig. 6
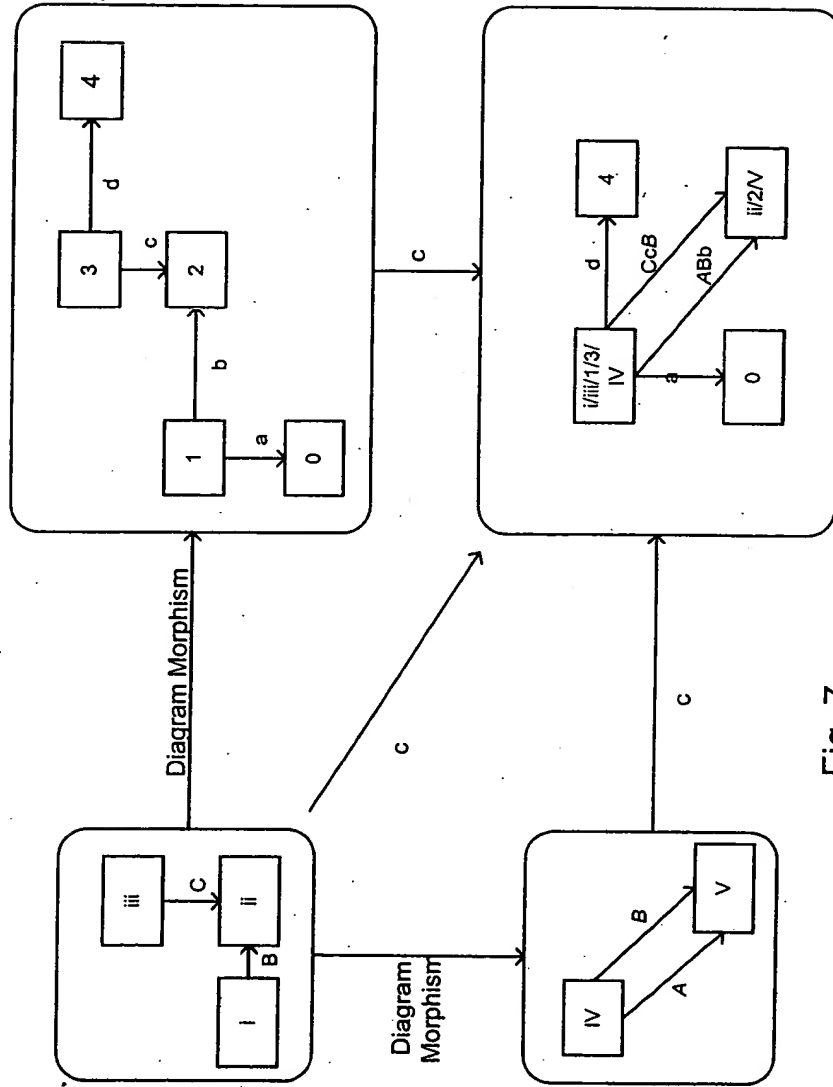
Fig. 7

Example of Taking the Colimit of Hereditary Diagrams

Bag-Diagram

- Import
- Refine (Lib)
- **Find Colimit of diagram**
...
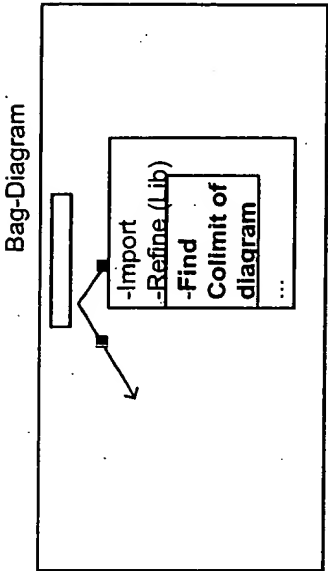
Example user interface showing a
hereditary diagram (interface for user
to indicate "find colimit" operation)
Fig. 8(b)
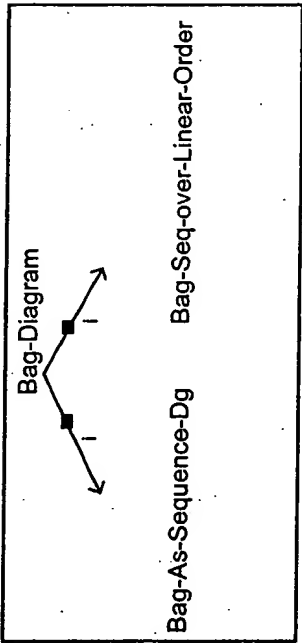
Bag-Diagram

Bag-Seq-over-Linear-Order

Bag-As-Sequence-Dg

Diagram-5

Example user interface showing a
hereditary diagram
Fig. 8(a)

Bag-Diagram

Bag-Seq-over-Linear-Order

Bag-As-Sequence-Dg
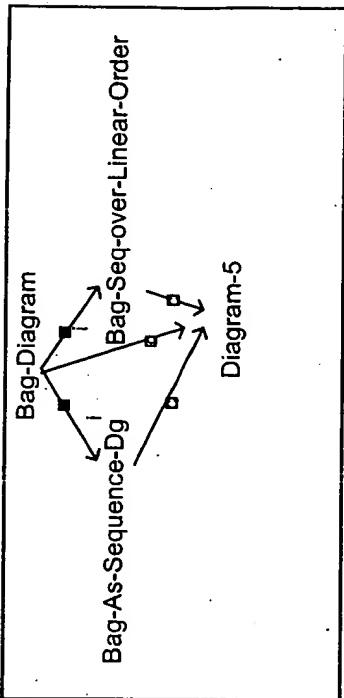
Example user interface showing a
hereditary diagram after the user
indicates a "find colimit" operation for
the hereditary diagram and the colimit
operation is performed

Fig. 8(c)

Specware Window 8

Bag-Diagram

Bag-Seq-Over-Linear-Order

Bag-as-Sequence-Dg

Hereditary diagram

Fig. 9(a)

Specware Window 26

Bag-Diagram

Bag-Seq-Over-Linord

c → Diagram-5

c

Bag-as-Sequence-Dg
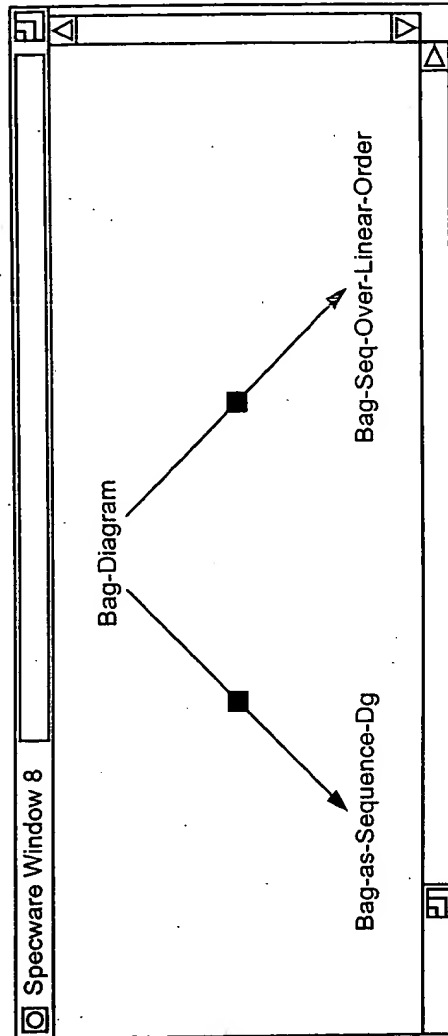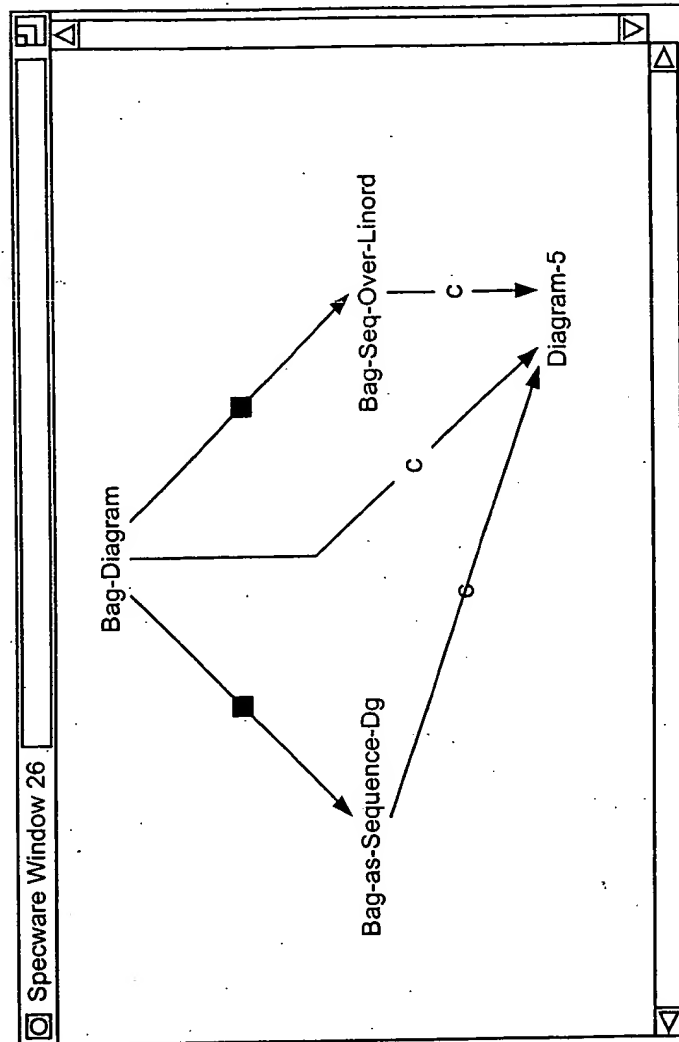
Hereditary diagram, including colimit

Fig. 9(b)

Specware W

Bag
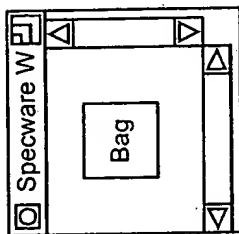
Bag diagram
(obtained by expanding node
Bag-Diagram
in Hereditary diagram)

Fig. 9(c)

Specware Window 13

Extended-Seq

d

Bag-As-Extended-Seq
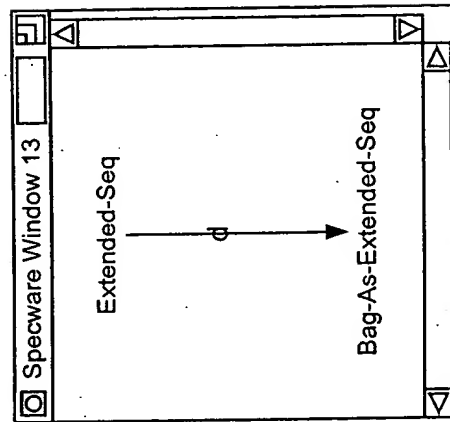
Bag-as-Sequence diagram
(obtained by expanding node
Bag-as-Sequence-diagram
in Hereditary diagram)

Fig. 9(d)

Initial: Spec Diagram BAG-TO-SEQ-SORTING-DG

Triv

Total-Order   Extended-Seq   Bag

                          c   c

                     Bag-And-Seq

              c

                     Bag-And-Seq-Conv

                          c

              Bag-Seq-Linord

                          d
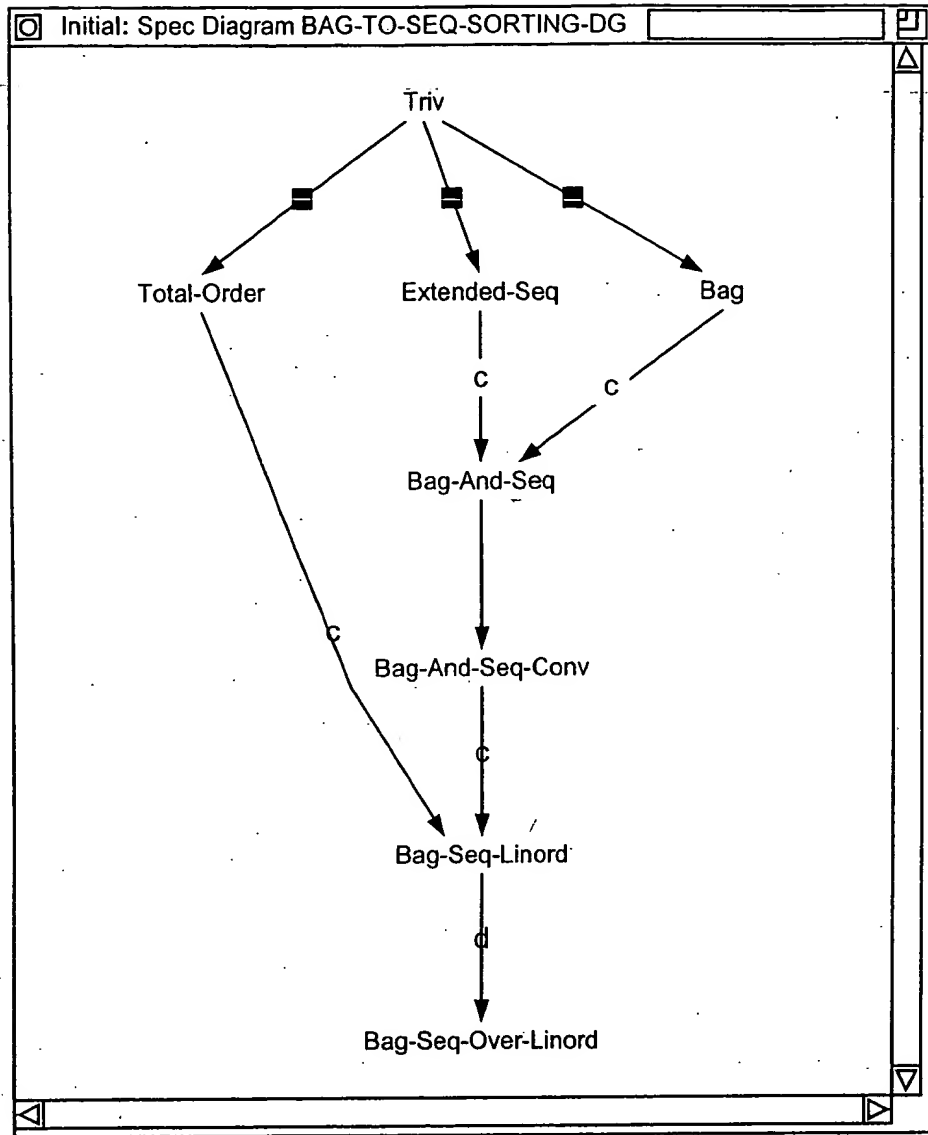
                     Bag-Seq-Over-Linord

Bag-Seq-over-Linear-Order diagram
(obtained by expanding node
Bag-Seq-over-Linear-Order-
diagram
in Hereditary diagram)

Fig. 9(e)
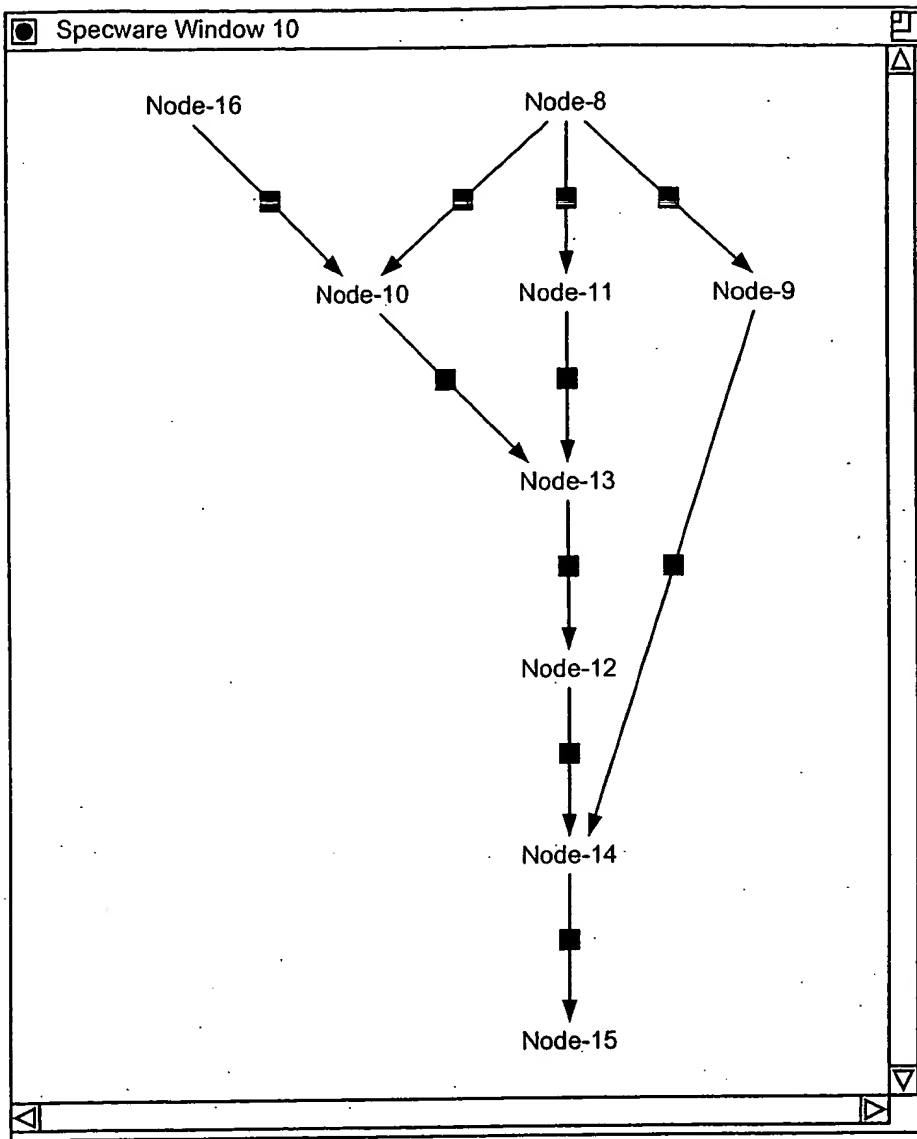
Specware Window 10
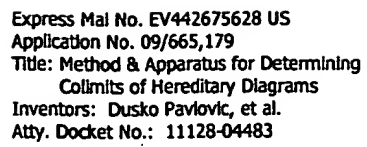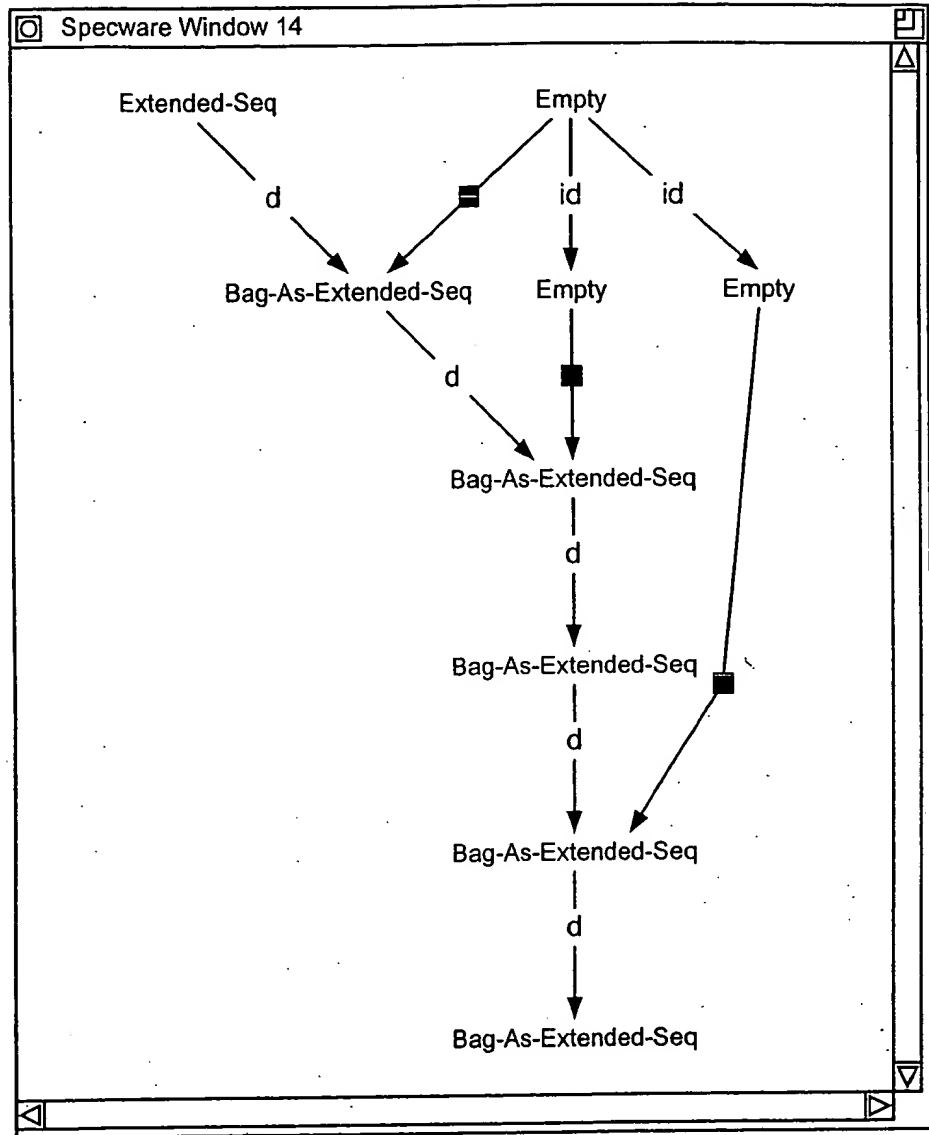
Node-16   Node-8

Node-10   Node-11   Node-9

Node-13

Node-12

Node-14

Node-15

Shape of colimit

Fig. 9(f)

Specware Window 18

Extended Bag diagram

Fig. 9(g)

Extended Bag-as-Sequence diagram

Fig. 9(h)

| ◻ | Specware Window 16 | | |

Empty   Triv

                               Bag   Extended-Seq   Total-Order

                                            Bag-And-Seq

                                            Bag-And-Seq-Conv

                                            Bag-Seq-Linord

                                       d

                                            Bag-Seq-Over-Linord

Extended Bag-Seq-over-Linear-Order diagram

Fig. 9(i)

Result: Spec-Diagram-5

Extended-Seq   Triv

d

Bag-As-Extended-Seq   Extended-Seq   Total-Order

Bag-And-Seq--Bag-As-Extended-Seq

d

Bag-And-Seq-Conv--Bag-As-Extended-Seq

Bag-Seq-Linord--Bag-As-Extended-Seq

d

Bag-Seq-Over-Linord--Bag-As-Extended-Seq
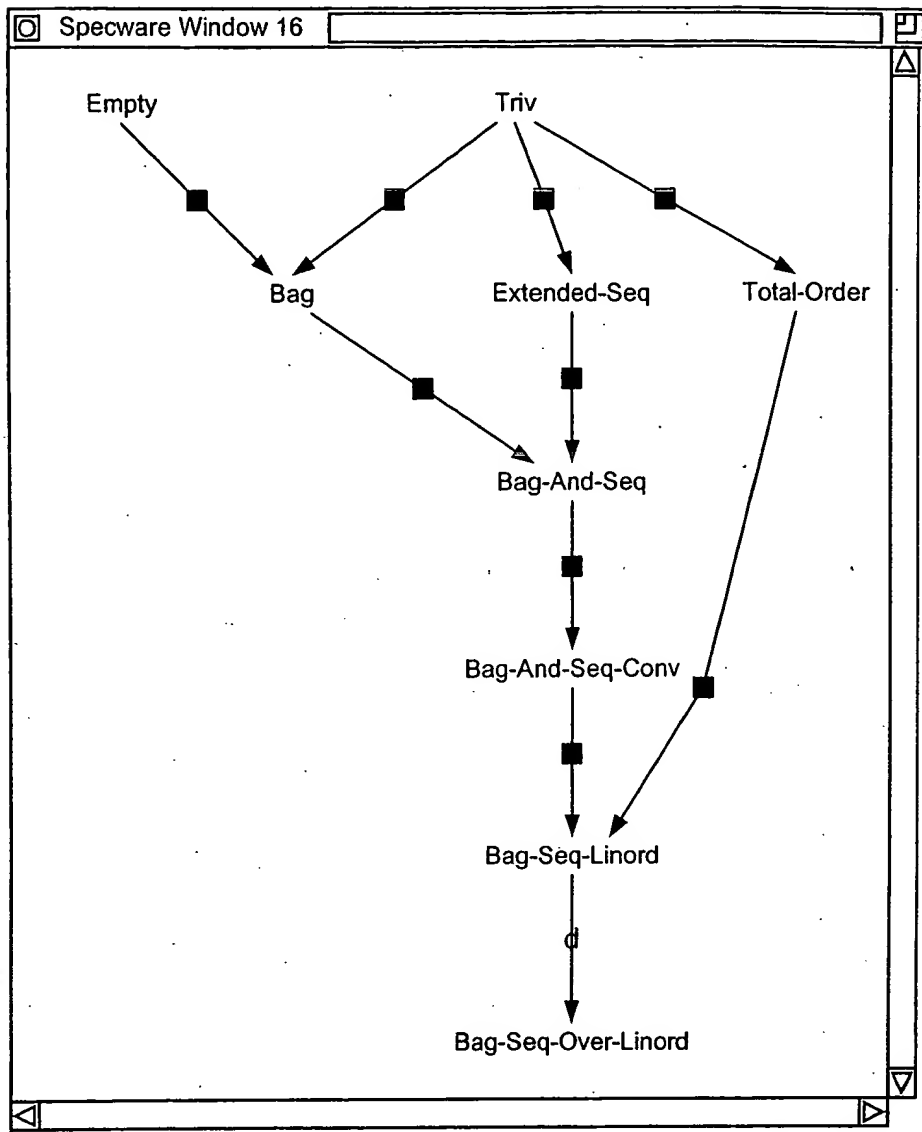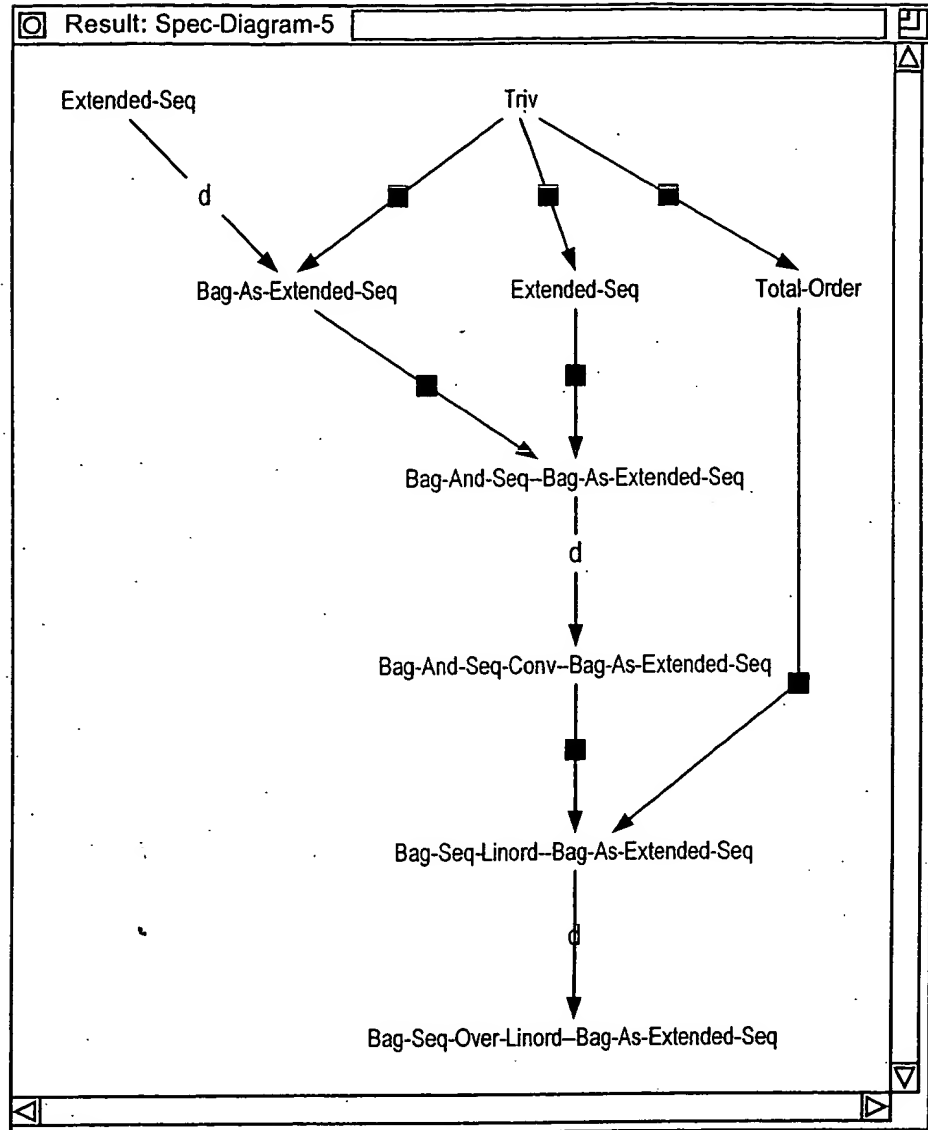
Colimit of Hereditary diagrams
(final result)

Fig. 9(j)

1002 User wants to take the colimit of a hereditary diagram (see Fig. 8(b))

1004 From the hereditary diagram, extract the shapes of its nodes and the functors between them to yield a diagram of shape categories. (see Fig. 11) Determine the colimit of the diagram of shape categories. The colimit of shape categories has the same shape as the colimit of diagrams that we are trying to determine (see Fig. 16)

I

1006 For each diagram of the hereditary diagram, determine its extension in accordance with the colimit shape obtained in the previous step. (This brings each of the diagrams to the same shape) (see Fig. 19)

II

1008 Take pointwise colimit of the extended diagrams to yield the desired colimit of the hereditary diagram. The morphisms of the colimit are uniquely determined by the universality of the pointwise colimits (Can take the pointwise colimit because the extended diagrams are all of the same shape) (See Fig. 22)

III

1010 Store newly determined colimit in a memory. Add name of newly determined colimit to display (User can view details of colimit if he wishes) (see Fig. 9)

Finding a Colimit of
Hereditary Diagrams

Fig. 10

Store in memory a set of the arcs in the diagram, i.e., arc_set := arcs in diagram

Store in memory a set of the nodes in the diagram, i.e., node_set := nodes in diagram

For each element in the set of arcs, associate with the arc (in memory) a source node (S) and a target node (T) in accordance with the connections of the diagram

Determine the morphisms (F_e) between the arcs of each source diagram and the arcs of each target diagram

Determine the morphisms (F_n) between the nodes of each source diagram and the nodes of each target diagram

for each diagram/ node in the hereditary diagram (3 in this example)

PART I: Extract the shapes and shape functors to yield a diagram of shape categories

Fig. 11

D1 - - - - F - - - -> D2

e

d1   SPEC   d2

A Shape Morphism

where d1 and d2 are diagrams,
F is a shape functor,
e is a natural transformation from d1 to
(d2 composed with F)
D1 and D2 are shape categories of
diagrams, and SPEC is the category
Spec

Fig. 12(b)

d1 $\xRightarrow{a1}$ d2

d1 $\xRightarrow{a2}$ d3
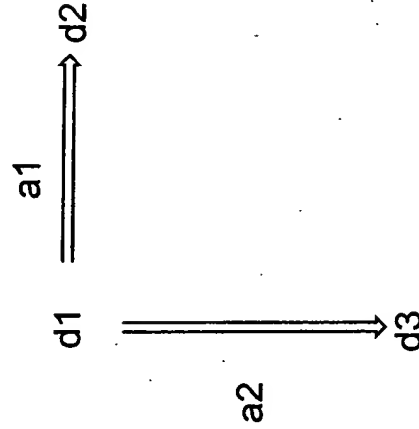
A Hereditary Diagram: Three
Diagrams and Two Arcs.

Each arc a1 and a2 represents a
shape morphism having 1) a
shape functor (such as F) and 2)
a natural shape transformation
(such as e: d1 --> d2)

Fig. 12(a)
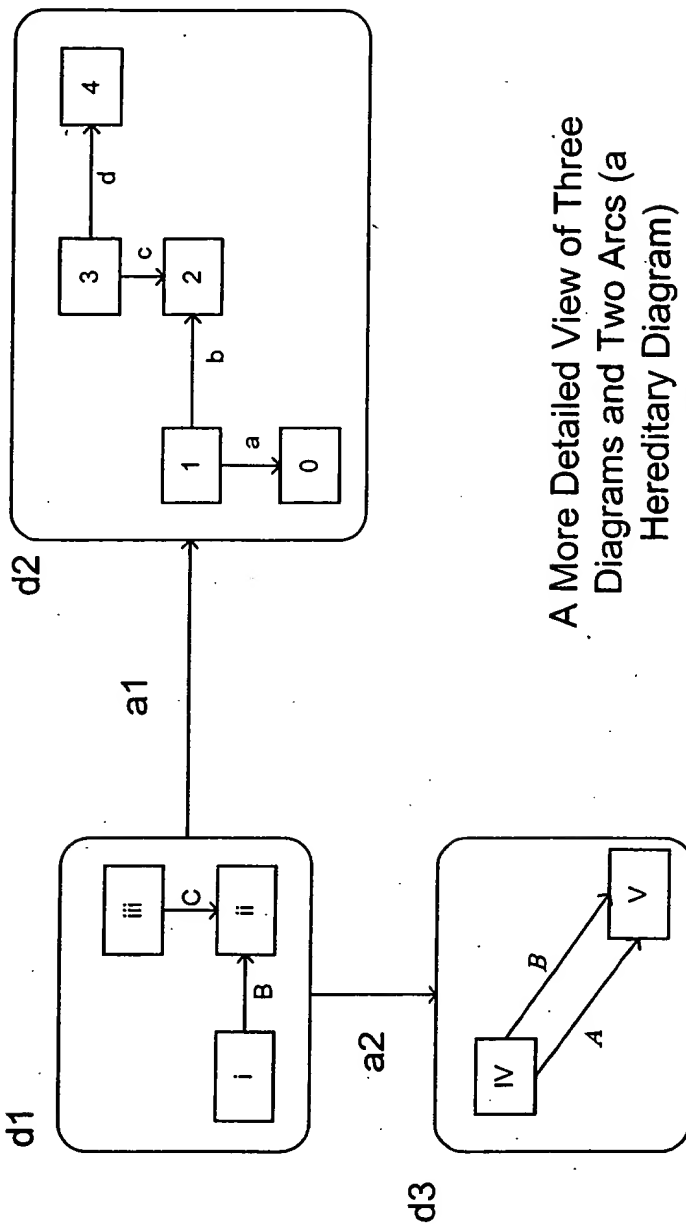
A More Detailed View of Three
Diagrams and Two Arcs (a
Hereditary Diagram)
Fig. 13

More Detailed View of Extracting the
Shapes and Shape Functors
(continued on Figs. 15(b)-15(d))
Fig. 15(a)
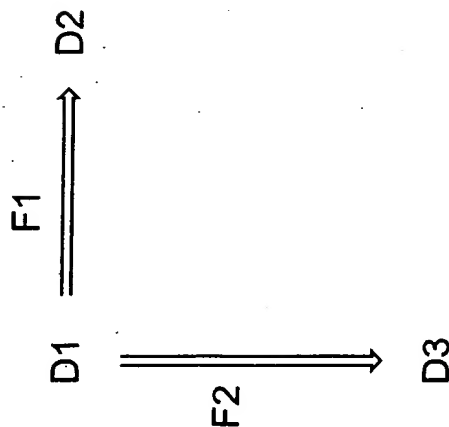
Extract the Shapes and
Shape Functors
(D1 is shape of diagram
d1, F1 is shape functor)
Fig. 14

Arcs: B -> A
      C -> B

Nodes: i -> IV
       ii -> V
       iii -> IV

Mapping for F2
Fig. 15(c)

Arcs: B -> b
      C -> c

Nodes: i -> 1
       ii -> 2
       iii -> 3

Mapping for F1
Fig. 15(b)

Source (S) and Target (T) Functions for Hereditary Diagrams

Fig. 15(d)

Arc	B	C	a	b	c	d	A	B
Source	i	iii	1	1	3	3	IV	IV
Target	ii	ii	0	2	2	4	V	V

Compute colimit of sets of nodes: Store in memory a disjoint union of all nodes (ignore arcs). Determine the equivalence relations identifying those nodes that are connected by some arc of the hereditary diagram. All nodes of the diagrams that fall in the same equivalence class are identified as a single node in the colimit
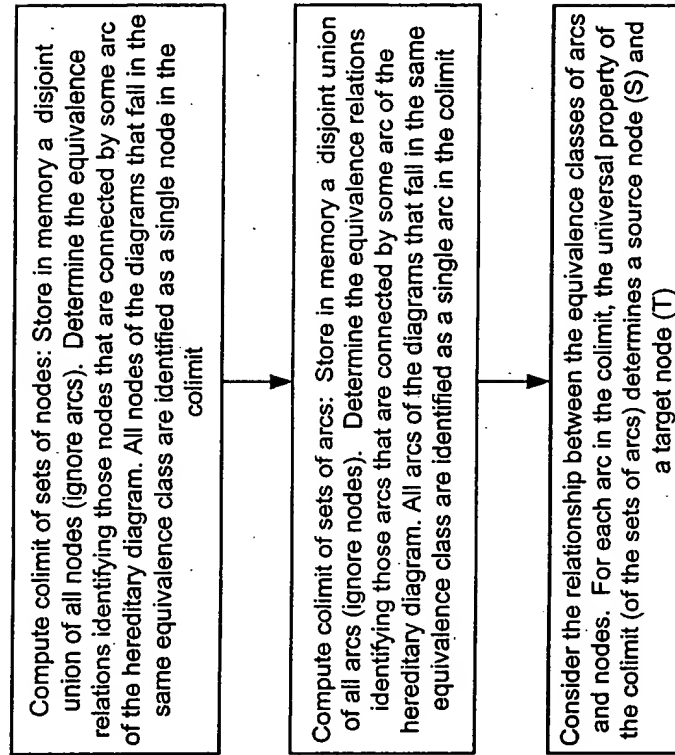
Compute colimit of sets of arcs: Store in memory a disjoint union of all arcs (ignore nodes). Determine the equivalence relations identifying those arcs that are connected by some arc of the hereditary diagram. All arcs of the diagrams that fall in the same equivalence class are identified as a single arc in the colimit

Consider the relationship between the equivalence classes of arcs and nodes. For each arc in the colimit, the universal property of the colimit (of the sets of arcs) determines a source node (S) and a target node (T)
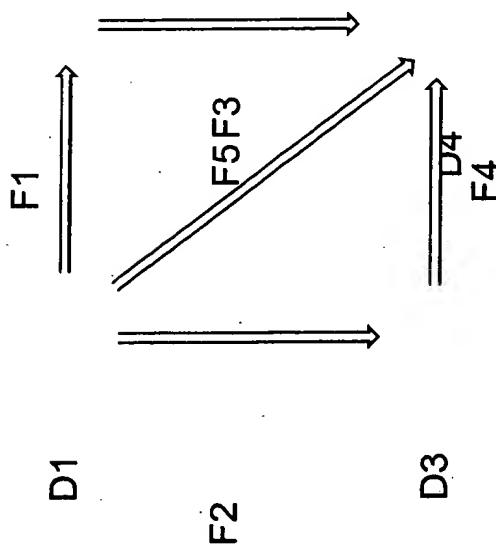
PART I: Determine the colimit of the diagram of shape categories.

Fig. 16
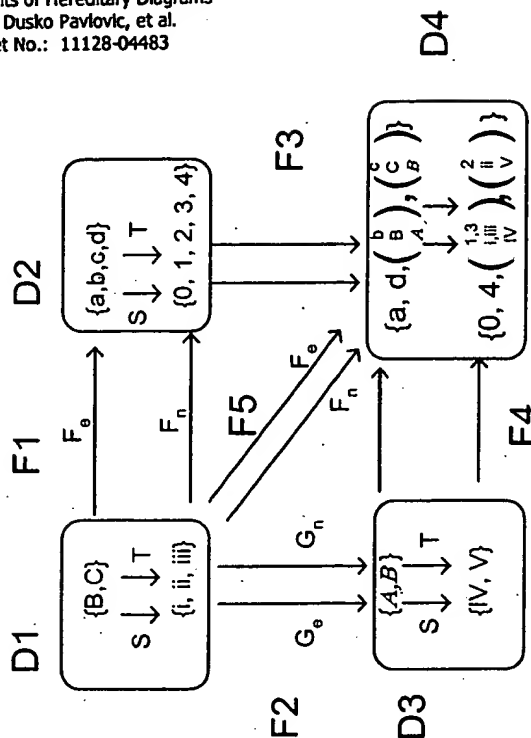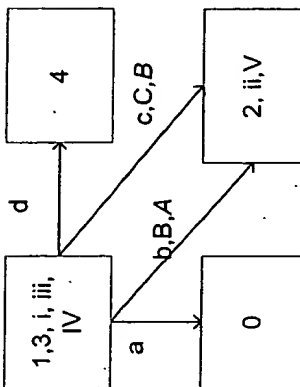
More Detailed View of Taking the Colimit
(See Figs 18(b)-(f))

Fig. 18(a)

More Detailed View of
Taking the Colimit
Fig. 17

The Colimit D4 of the Shape Diagrams
Fig. 18(c)

Source (S) and Target (T) Functions for Shape Colimit D4
Fig. 18(b)

Arc	a	d	b A A	c C B
Source	1,3 i,iii IV	1,3 i,iii IV	1,3 i,iii IV	1,3 i,iii IV
Target	0	4	2 ii V	2 ii V

Arcs: a -> a
      d -> d
      b -> b,B,A
      c -> c,C,B

Nodes: 0 -> 0
        1 -> 1,3, i,iii, IV
        2 -> 2, ii,V
        3 -> 1,3, i,iii, IV
        4 -> 4

Mapping for F3
Fig. 18(d)

Arcs: A -> b,B,A
      B -> c,C,B
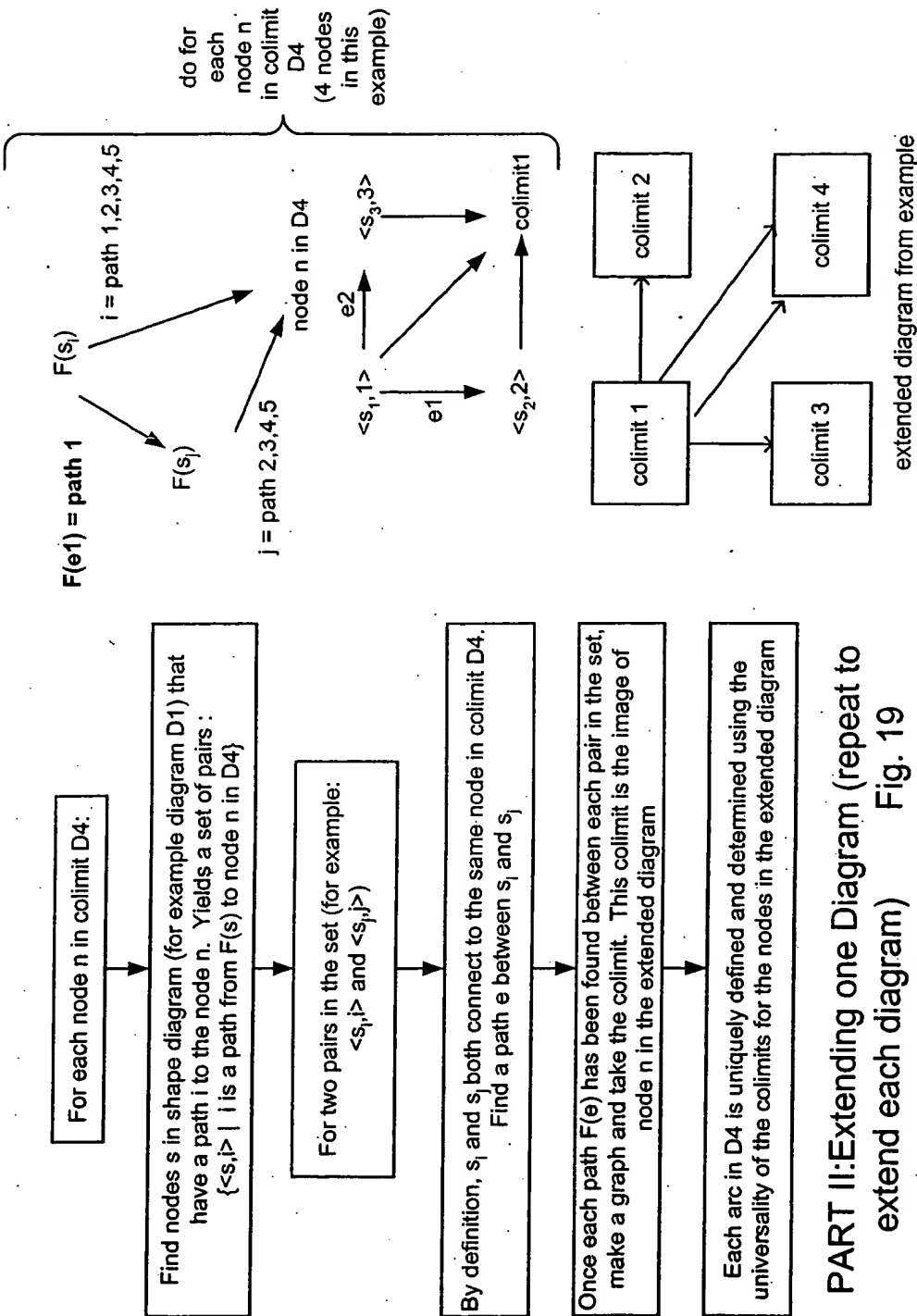
Nodes: IV -> 1,3, i, iii, IV
        V -> 2, ii,V

Mapping for F4
Fig. 18(e)

Arcs: B -> b,B,A
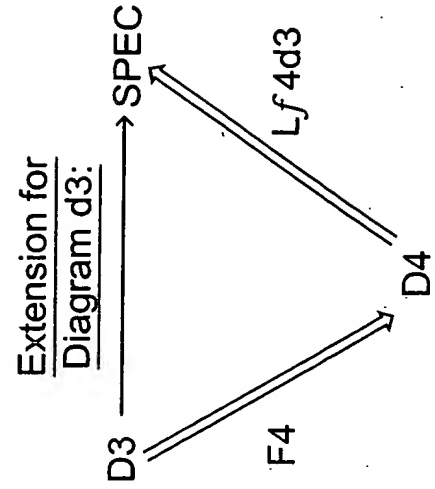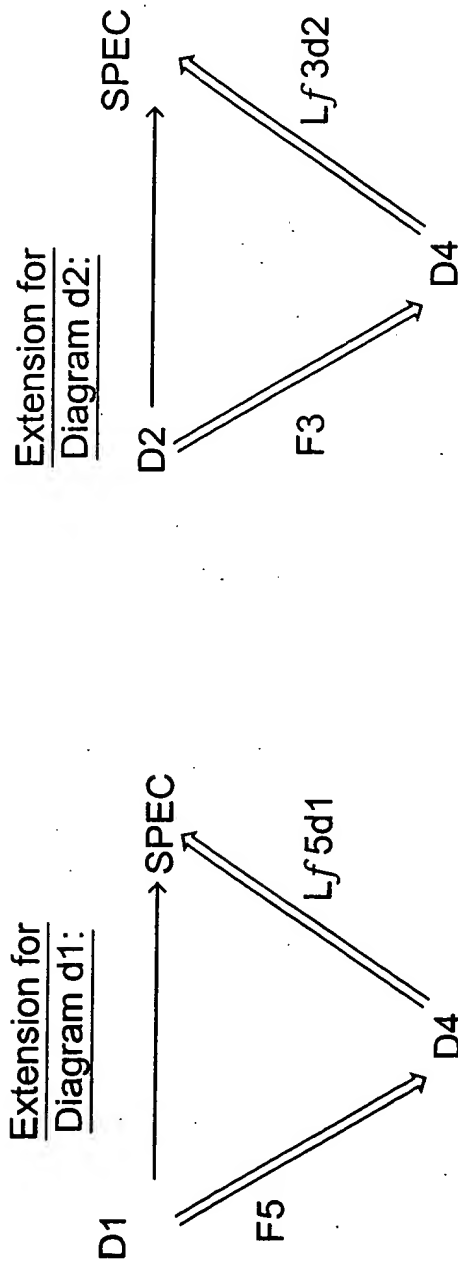   C -> c,C,B

Nodes: i -> 1,3, i, iii, IV
        ii -> 2, ii,V
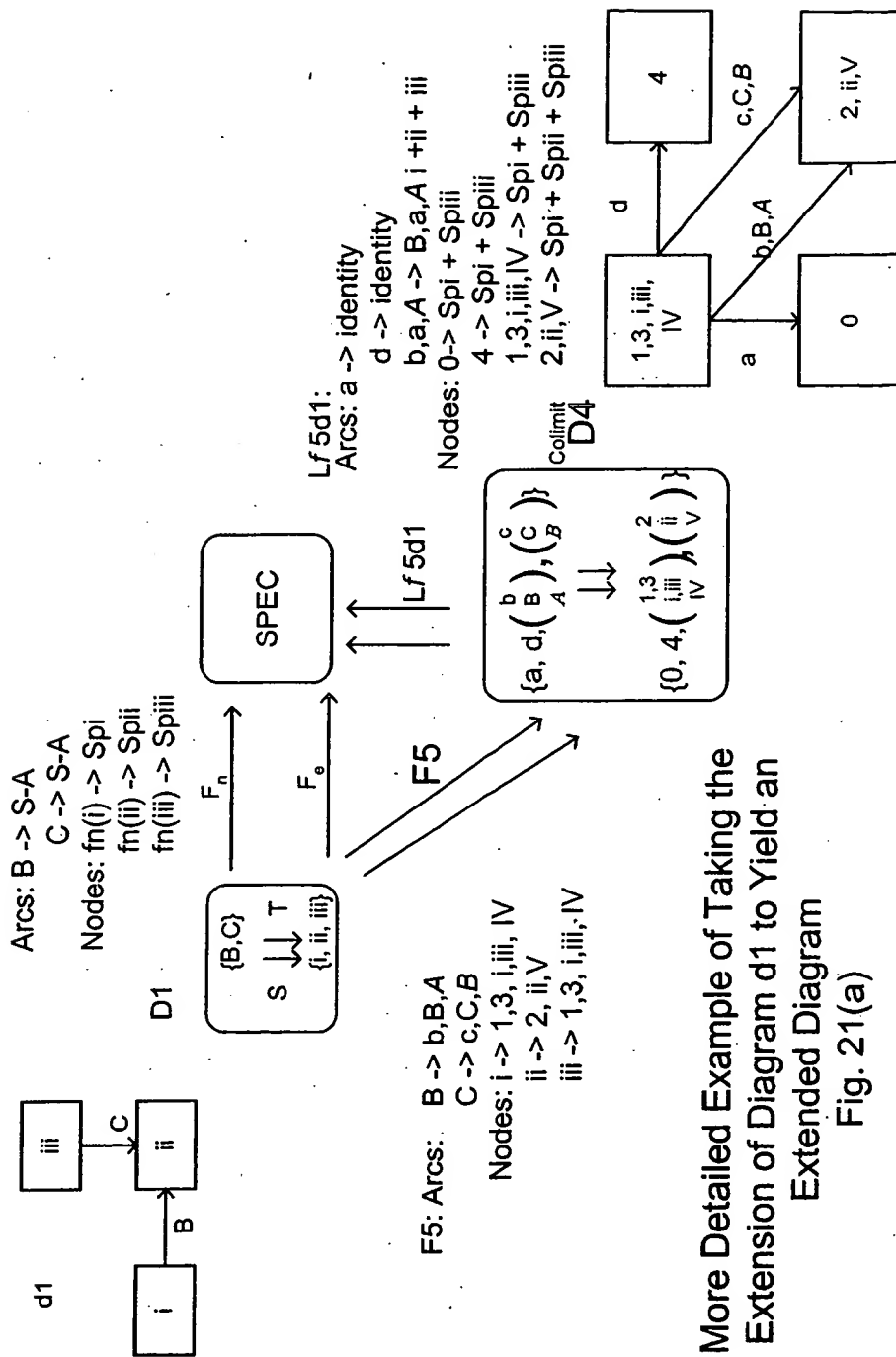   iii -> 1,3, i, iii, IV

Mapping for F5
Fig. 18(f)

do for each node n in colimit D4 (4 nodes in this example)

$F(e1) = $ path 1

$F(s_i)$

$i = $ path 1,2,3,4,5

$F(s_j)$

$j = $ path 2,3,4,5

node n in D4

$<s_3,3>$

$<s_1,1>$

e2

e1

$<s_2,2>$

colimit1

colimit 2

colimit 1

colimit 4

colimit 3

extended diagram from example

For each node n in colimit D4.

Find nodes s in shape diagram (for example diagram D1) that have a path i to the node n. Yields a set of pairs : {$<s,i>$ | i is a path from F(s) to node n in D4}

For two pairs in the set (for example: $<s_i,i>$ and $<s_j,j>$)

By definition, s_i and s_j both connect to the same node in colimit D4. Find a path e between s_i and s_j

Once each path F(e) has been found between each pair in the set, make a graph and take the colimit. This colimit is the image of node n in the extended diagram

Each arc in D4 is uniquely defined and determined using the universality of the colimits for the nodes in the extended diagram

PART II: Extending one Diagram (repeat to extend each diagram)   Fig. 19

Extension for
Diagram d1:

$Lf5d1$

D1

F5

D4

SPEC

Extension for
Diagram d2:

$Lf3d2$

D2

F3

D4

SPEC

Extension for
Diagram d3:

$Lf4d3$

D3

F4

D4

SPEC

Example of Taking the
Extension of Each Node of the
Hereditary Diagram
Fig 20

Arcs: B -> S-A
C -> S-A
Nodes: fn(i) -> Spi
fn(ii) -> Spii
fn(iii) -> Spiii

F_n

F_e

$\{B,C\}$
S $\Downarrow$ T
$\{i, ii, iii\}$

D1

F5

SPEC

Lf 5d1

Lf 5d1:
Arcs: a -> identity
d -> identity
b,a,A -> B,a,A i +ii + iii
Nodes: 0-> Spi + Spii
4 -> Spi + Spiii
1,3,i,iii,IV -> Spi + Spii + Spiii
2,ii,V -> Spi + Spii + Spiii

Colimit
D4

$\{a, d, \binom{b}{B}_A, \binom{c}{C}_B\}$
$\Downarrow$
$\{0, 4, \binom{1,3}{i,iii}_{IV}, \binom{2}{ii}_V\}$

F5: Arcs: B -> b,B,A
C -> c,C,B
Nodes: i -> 1,3, i,iii, IV
ii -> 2, ii,V
iii -> 1,3, i,iii, IV

d1

iii
C
ii
B
i

4
d
c,C,B
1,3, i,iii,
IV
b,B,A
2, ii,V
a
0

More Detailed Example of Taking the
Extension of Diagram d1 to Yield an
Extended Diagram
Fig. 21(a)
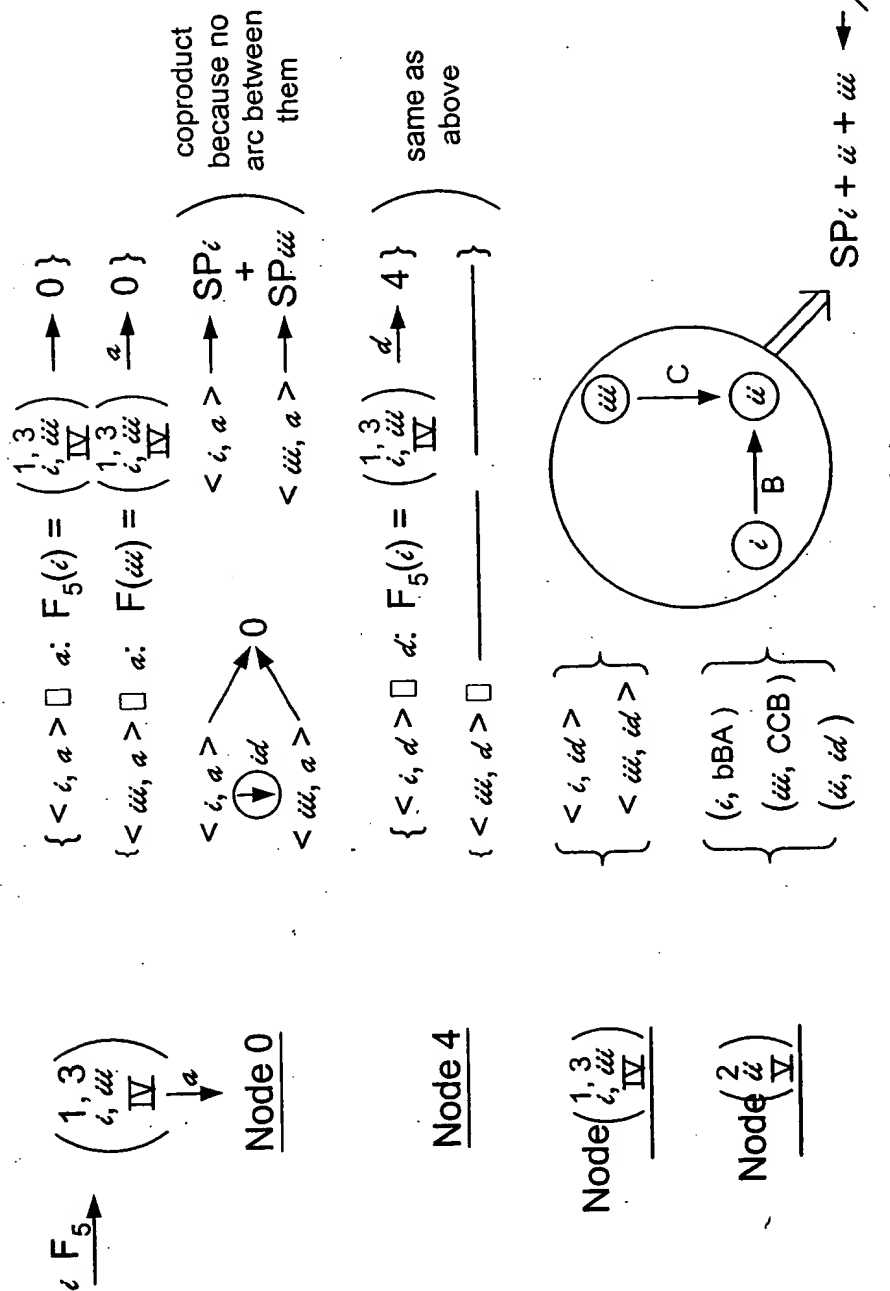
Fig. 21(b)

Example of Diagram Extension.
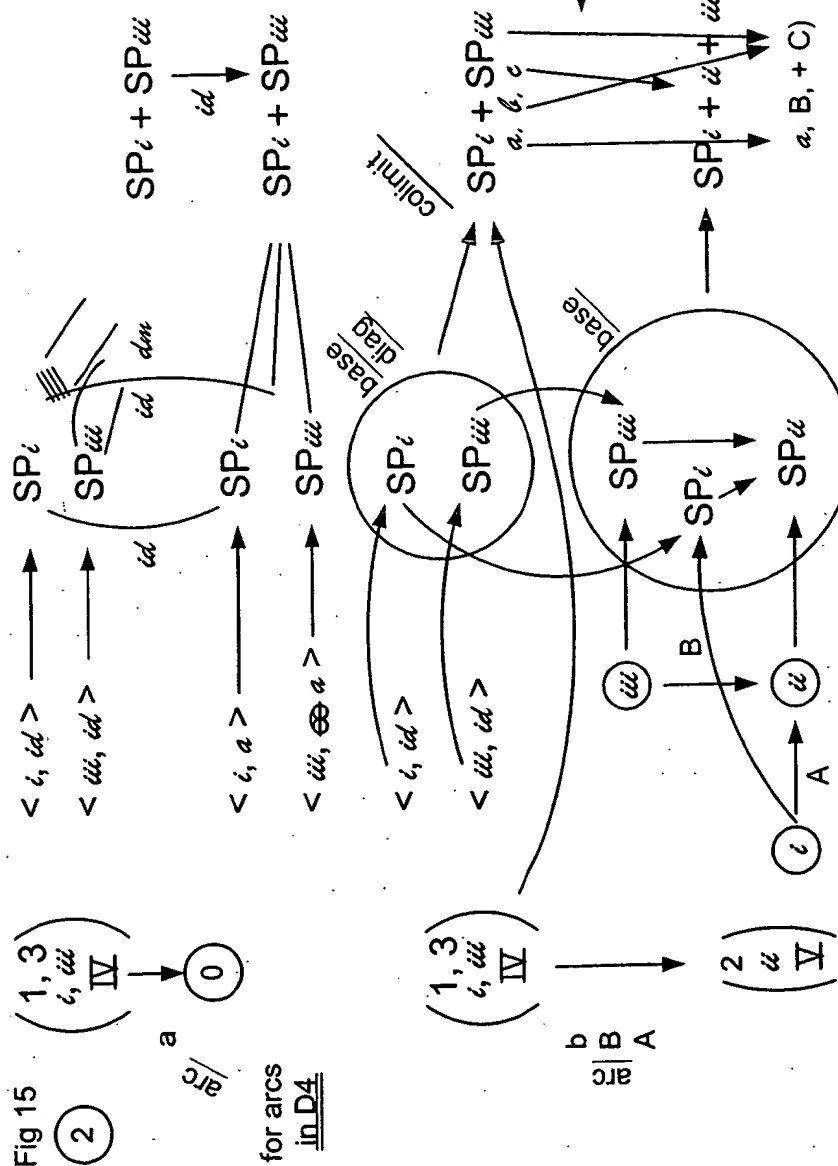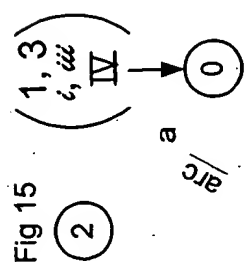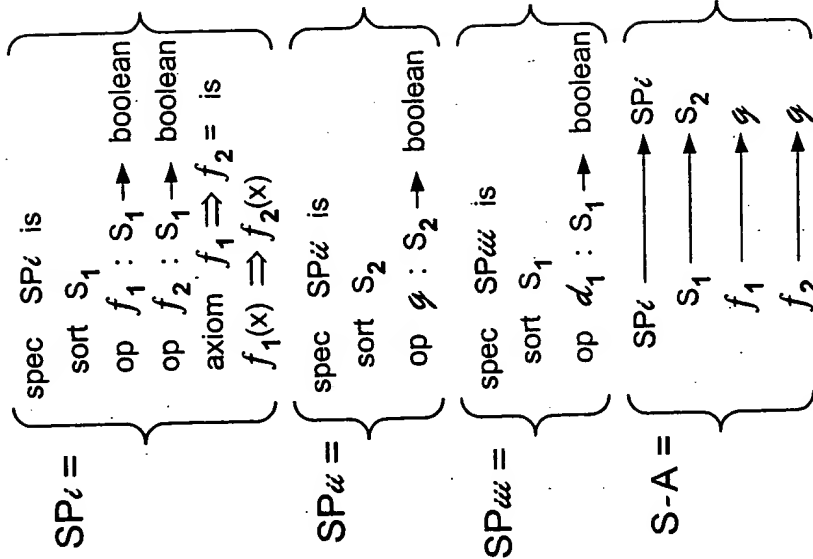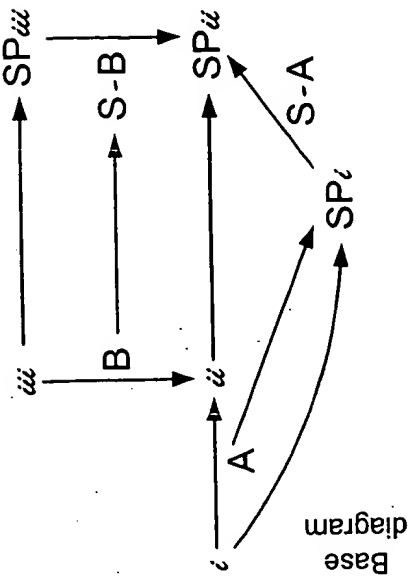
Fig. 21(c)

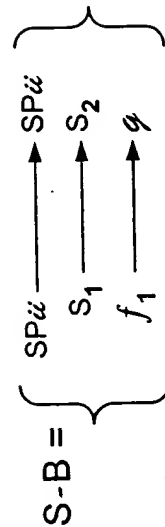Example of Diagram Extension (cont)

$$SP\ddot{u} \longrightarrow SP\dot{u}$$

Base diagram

$$S\text{-}B =
\begin{cases}
SP\ddot{u} \longrightarrow SP\dot{u} \\
S_1 \longrightarrow S_2 \\
f_1 \longrightarrow g
\end{cases}$$

Fig. 21(d)

$$SP\dot{\imath} =
\begin{cases}
\text{spec} \quad SP\dot{\imath} \text{ is} \\
\text{sort} \quad S_1 \\
\text{op} \quad f_1 : S_1 \to \text{boolean} \\
\text{op} \quad f_2 : S_1 \to \text{boolean} \\
\text{axiom} \quad f_1 \rightrightarrows f_2 = \text{is} \\
\qquad f_1(x) \rightrightarrows f_2(x)
\end{cases}$$

$$SP\ddot{u} =
\begin{cases}
\text{spec} \quad SP\ddot{u} \text{ is} \\
\text{sort} \quad S_2 \\
\text{op} \quad g : S_2 \to \text{boolean}
\end{cases}$$

$$SP\dddot{u} =
\begin{cases}
\text{spec} \quad SP\dddot{u} \text{ is} \\
\text{sort} \quad S_1 \\
\text{op} \quad d_1 : S_1 \to \text{boolean}
\end{cases}$$

$$S\text{-}A =
\begin{cases}
SP\dot{\imath} \longrightarrow SP\ddot{u} \\
S_1 \longrightarrow S_2 \\
f_1 \longrightarrow g \\
f_2 \longrightarrow g
\end{cases}$$
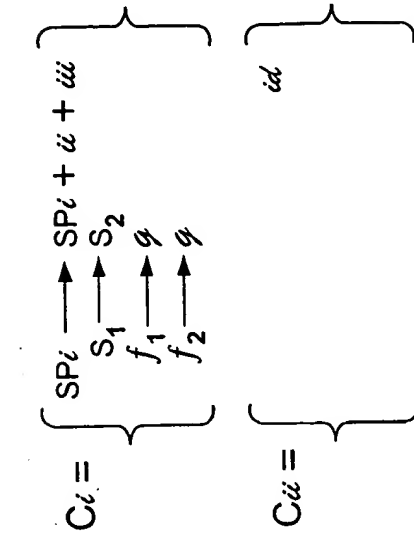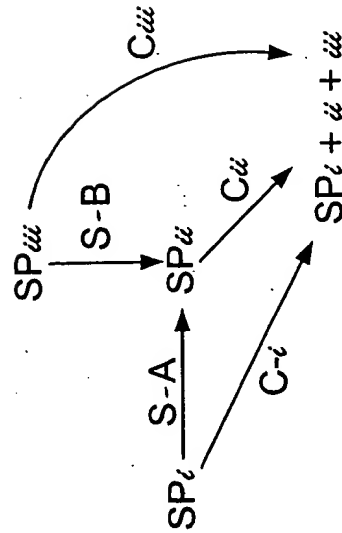
Example of Diagram Extension (cont)

Example of Diagram Extension (cont)   **Fig. 21(e)**

$$SP i + SP iii = \left\{ \begin{array}{l} \text{sort} \quad SP i \quad S_1 \\ \text{sort} \quad SP iii \text{-} S_1 \\ \text{op} \quad SP i_1 \, f_1 \; : \; SP i \text{-} S_1 \longrightarrow \text{boolean} \\ \text{op} \quad SP iii \, f_1 \; : \; SP iii \text{-} S_1 \longrightarrow \text{boolean} \\ \text{op} \quad SP i \, f_2 \quad : \; SP i \text{-} S1 \longrightarrow \text{boolean} \end{array} \right\}$$

$$\downarrow \begin{array}{l} b \\ B \; i + ii + iii \\ A \end{array}$$

$$SP i + ii + iii = \left\{ \quad \text{See Fig. 21(e)} \quad \right\}$$

$$\begin{array}{l} b \\ B \; i + ii + iii = \\ A \end{array} \left\{ \begin{array}{l} SP i \cdot S_1 \longrightarrow S_2 \\ SP iii \cdot S_1 \longrightarrow S_2 \\ SP i \cdot f_1 \longrightarrow g \\ SP iii \cdot f_1 \longrightarrow g \\ SP iii \cdot f_2 \longrightarrow g \end{array} \right\}$$
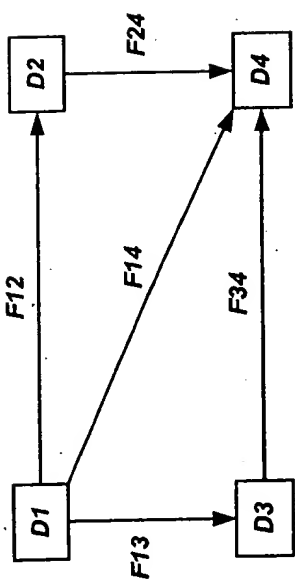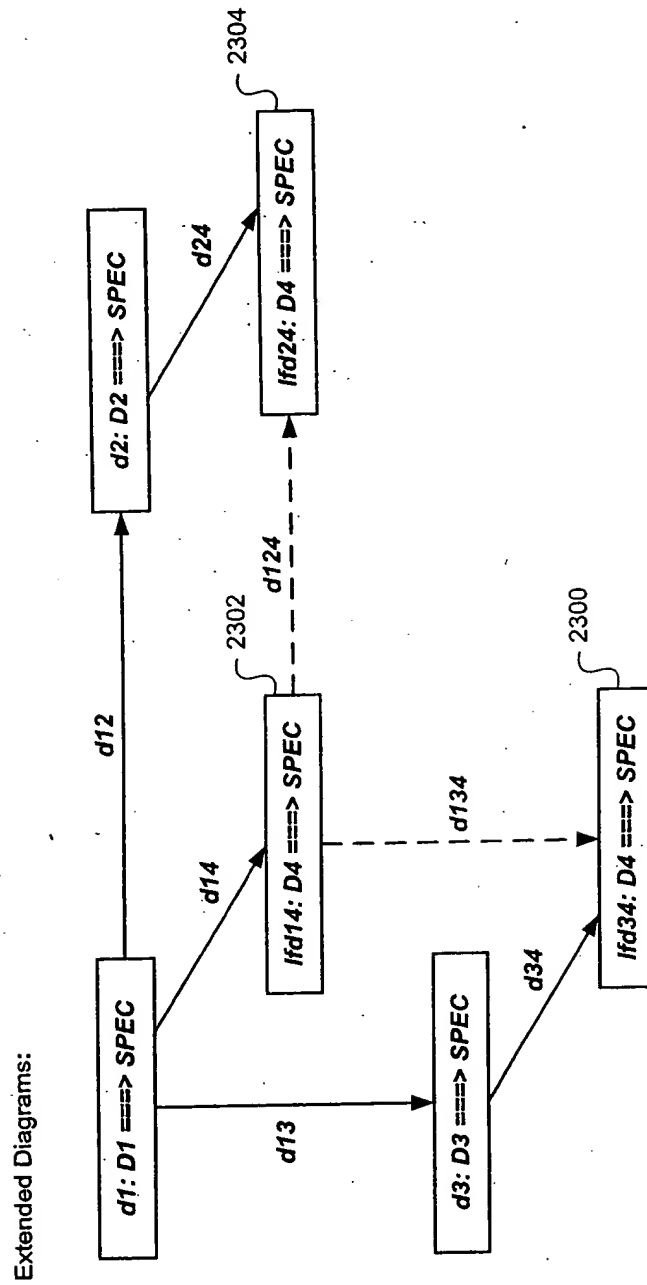
Example of Diagram Extension (cont)

Fig. 21(f)

After finishing the extension for each diagrams, let us use the following example:

Original diagrams:
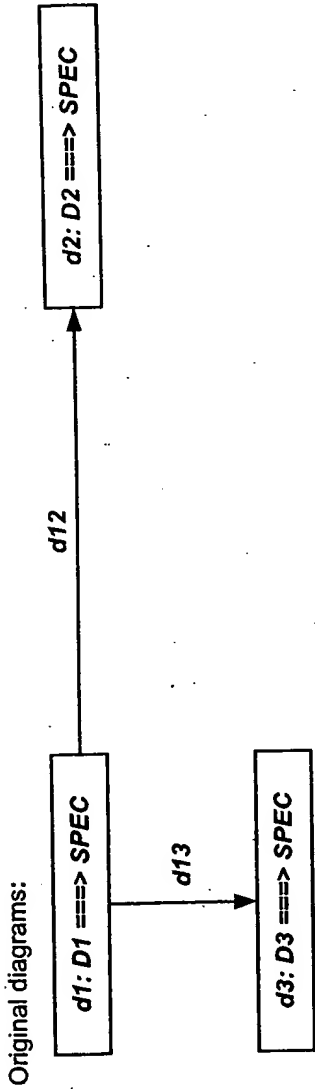
Its underlying shape categories, shape functors and the colimit are:
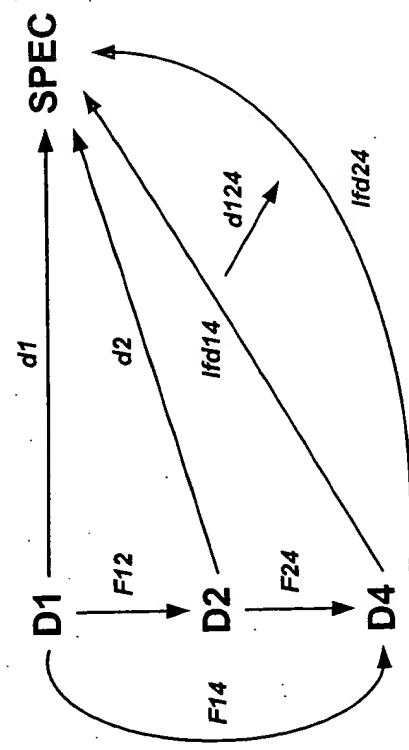

Part III

Fig. 22

Extended Diagrams:

Fig. 23

The last algorithm step we are missing for constructing the diagram colimits is the diagram morphisms between extended diagrams. For example, the diagram morphism d124 and d134 (dotted lined arrows in above figure) are the ones needed.

Suppose lfd14 and lfd24 are two extensions of d1 and d2, given the colimit of the shape categories as D4. We would have the following picture.

Express Mai No. EV442675628 US
Application No. 09/665,179
Title: Method & Apparatus for Determining
         Colimits of Hereditary Diagrams
Inventors: Dusko Pavlovic, et al.
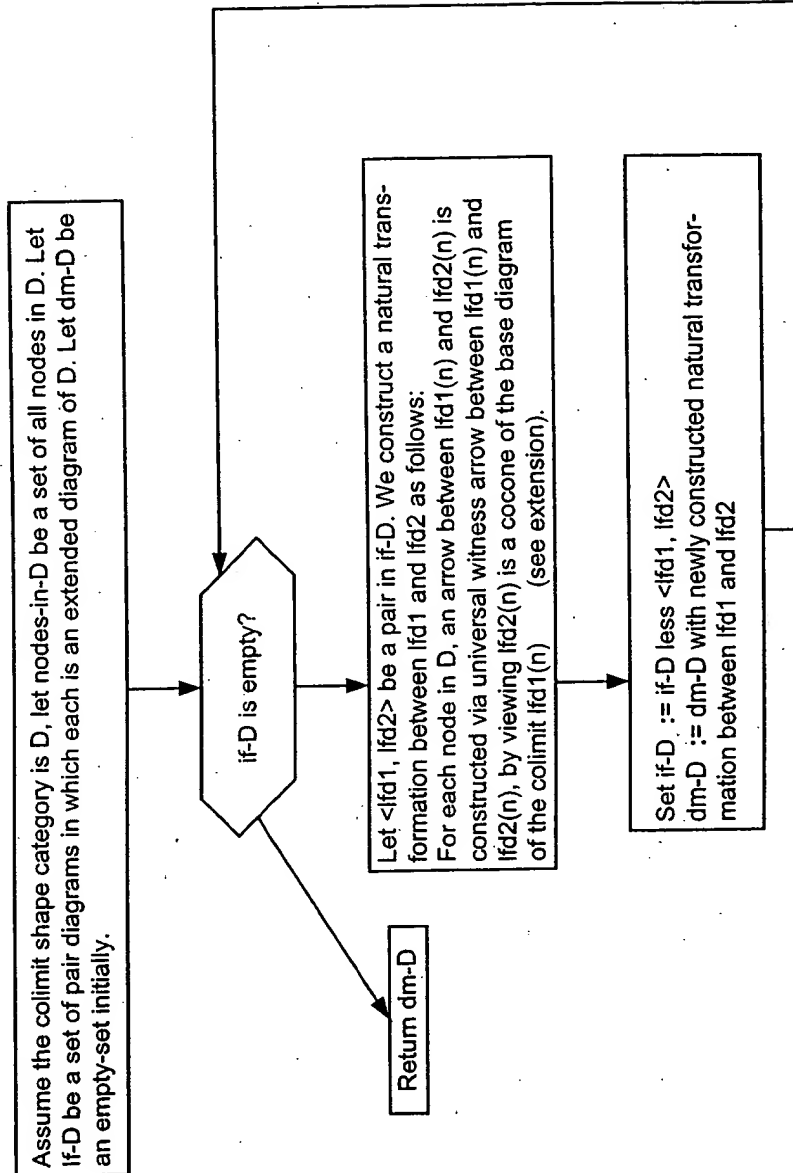Atty. Docket No.: 11128-04483

Fig. 24

A morphism between lfd14 and lfd24 is a natural transformation which maps each node of D4 to an arrow in SPEC. We ?? ????
the universal construction of witness arrows.

For any node ni in D4, we have F14(ni) = F12 o F24(ni). Let Sp1ni and Sp2ni two shape categories used for constructing a mapping
for ni in its extension of d1 and d2, respectively, then we can have a shape function between Sp1ni and Sp2ni (inclusion, basi-
cally). That induces a diagram morphism between the base diagrams for the target of ni in lfd14 and lfd24, respectively. By imposing
that diagram morphism and cocone morphism, we can get an unique arrow between lfd14(ni) and lfd24(ni). Repeating this process we
construct a natural transformation between lfd14 and lfd24. Similarly, we can do this for any two extended diagrams.

The following flowchart is the algorithm for constructing a diagram morphism between two extended diagrams.

Assume the colimit shape category is D, let nodes-in-D be a set of all nodes in D. Let
If-D be a set of pair diagrams in which each is an extended diagram of D. Let dm-D be
an empty-set initially.

if-D is empty?

Return dm-D

Let <lfd1, lfd2> be a pair in if-D. We construct a natural trans-
formation between lfd1 and lfd2 as follows:
For each node in D, an arrow between lfd1(n) and lfd2(n) is
constructed via universal witness arrow between lfd1(n) and
lfd2(n), by viewing lfd2(n) is a cocone of the base diagram
of the colimit lfd1(n)   (see extension).

Set if-D := if-D less <lfd1, lfd2>
dm-D := dm-D with newly constructed natural transfor-
mation between lfd1 and lfd2

Fig. 25

The final step is to complete the colimit of the extended diagrams. The colimit is determined by computing the pointwise colimits over corresponding nodes in the extended diagrams. The morphisms are computed uniquely using universality of the pointwise colimits.

Extended diagram of d11

Extended diagram d2

Extended diagram d3

Colimit of Diagrams
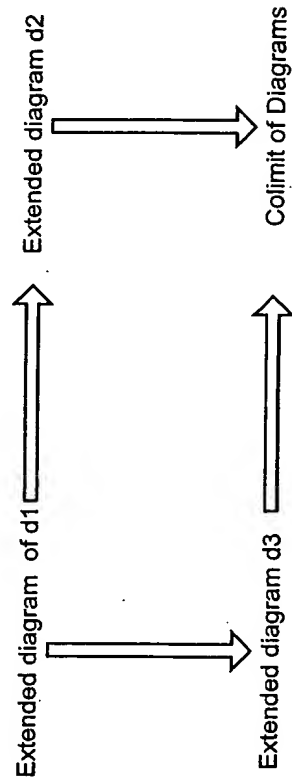
Taking Pointwise Colimit of Extended Diagrams
(Can be done, since extended diagrams are all the same shape)

Fig. 26

Shape Morphism Graph

Diagram Category Pair	Arc

Arc

Shape functor (Fe)	Natural Transformation (Fn)

Hereditary Diagram

Arc; source and target diagrams	...	Arc; source and target diagrams	Total number of Arcs

Diagram

Arc; source and target nodes	...	Arc; source and target nodes	Total number of Arcs

Examples of Data Structures used in the Example Implementation

Fig. 27

Fig. 28

Fig. 29